Amendments to the Drawings:

Included in the amendment are an "Annotated Sheet Showing Changes" and a "Replacement Sheet" for Figs. 1 and 5A. In Fig. 1, the text "MRF including a processor state register" was added to element 125 per the Examiner's requirement in the parent case under 37 C.F.R. §1.83(a) that this claimed element be illustrated in the drawings. Fig. 5A has been amended to disconnect the instruction decode and iVLIW control element found in element 553 from the Bcast data bus. Fig. 5A also has been amended to connect the instruction decode and iVLIW control element shown in element 553 to the C-bit instruction bus and to be consistent with the connections shown for similar elements; element 551 and elements 555 and 557 of Fig. 5B. The amendment to Fig. 5A is made to correct a typographical drawing error.

Remarks

The present amendment responds to the Official Action dated July 22, 2004. The Official Action objected to claims 14, 16, 19, 24, and 26 for informalities. Claim 19 was rejected under 35 U.S.C §112, second paragraph, as being indefinite for lack of antecedent basis. The Official Action rejected claims 14, 15 and 18-23 under 35 U.S.C. §103(a) based on Panwar et al. U.S. Patent No. 5,890,008 (Panwar) in view of Gordon et al. U.S. Patent No. 4,135,247 (Gordon). Claims 16 and 17 were rejected under 35 U.S.C.§103(a) based on Panwar in view of Gordon and further in view of Dowling U.S. Patent No. 6,128,728 (Dowling). Claims 24-27 and 29 were rejected under 35 U.S.C. §103(a) based on Panwar in view of Gordon and further in view of Keckler et al. U.S. Patent No. 5,574,939 (Keckler). Claim 28 was rejected under 35 U.S.C. §103(a) based on Panwar in view of Gordon and Keckler, and further in view of Bapst et al. U.S. Patent No. 6,327,650 (Bapst). These grounds of rejection are addressed below.

Claims 14, 16, 19, 24, and 26 have been amended to be more clear and distinct. More particularly, claims 14, 16, 19, 24, and 26 have been amended to address the informality objections as discussed below. Claim 19 has also been amended to provide an antecedent for the term "the second operating context." Claims 14-29 are presently pending.

Specification Amendment

The paragraph beginning at page 9, line 18 has been amended to add the term "processor state register" which is utilized in the claims. The processor state register is one example of a

control register. It should be noted that the term "processor state register" has been used in the original claims and is thus part of the original disclosure. Thus, no new matter has been added.

Drawing Amendments to Fig. 1 and Fig. 5A

In response to the requirements under 37 C.F.R. §1.83(a) in the parent case, Fig. 1 has been amended so that a claim element, the "processor state register", is illustrated in the drawings. The amendment to Fig. 5A is made to correct a typographical drawing error to be consistent with the connections shown for similar elements; element 551 and elements 555 and 557 of Fig. 5B.

Informality Objections to Claims 14, 16, 19, 24, and 26

The Examiner is thanked for his careful reading of these claims and proposed suggestions for improving the clarity of said claims. Claim 7 has been amended to address the antecedent basis objection. Claim 14 has been amended to remove the term "at least" when this term modified the term "a first state." Claim 16 has been amended to add the phrase "the data contents of" in order to clarify that data contents are stored in the first set of registers. Claim 19 has been amended to replace the term "the second software task" with the term "a second software task," to address the antecedent basis objection. Claim 24 has been amended to correct the duplicate use of the article "a." Claim 26 has been amended to use the article "an" instead of the article "a" when specifying an SP resource.

Section 112, Second Paragraph Rejection

Claim 19 has been amended to address the antecedent basis objection by replacing the phrase "the second operating context" with the phrase "a second operating context."

The Art Rejections

As addressed in greater detail below, Panwar, Gordon, Dowling, Keckler and Bapst do not support the Official Action's reading of them and the rejections based thereupon should be reconsidered and withdrawn. Further, the Applicant does not acquiesce in the analysis of Panwar, Gordon, Dowling, Keckler and Bapst made by the Official Action and respectfully traverses the Official Action's analysis underlying its rejections.

Claims 14, 15 and 18-23 were rejected under 35 U.S.C. §103(a) based on Panwar in view of Gordon. Panwar describes a method for dynamically reconfiguring a processor which places the processor in a first configuration having a first number (m) of virtual processors while the coded instructions comprise instructions from a number (m) threads or processes. The instructions in each of the m threads are executed on one of the m virtual processors using execution resources at least some of which are shared among the m virtual processors.

In the case that the coded instructions comprise instructions from a number (n) threads, the processor is placed in a second configuration having a second number (n) virtual processors. The instructions are executed in each of the n virtual processors using execution resources at least some of which are shared among the n virtual processors. Panwar, col. 4, lines 15-27. There is one state machine for each virtual processor. Panwar, col. 7, lines 22-24 and Fig. 3.

Panwar's approach involves a single physical processor which executes a plurality of virtual processors depending on the needs of a software application.

In contrast, the present invention addresses an advantageous array processor which can execute at least two software programs which were designed to execute on two physically different processor configurations. For example, one software application which has only sequential operations may be designed for executing on an array processor configuration containing only one processing element(PE) while a second software application which has many parallel operations may be designed for executing on an array processor configuration containing at least two processing elements. To operate an array processor in multiple contexts, the present invention advantageously utilizes a context status bit (CSB) which is stored in a processor state register to distinguish multiple physical configurations. Claim 14, as presently amended, reads as follows:

> An array processor comprising:
> a physical configuration of at least two processing elements; and
> a processor state register storing a context status bit (CSB), the CSB having a first state and a second state, each processing element operating to detect the state of the CSB,
> the array processor upon detection of the first state of the CSB operating in a first operating context adapted for processing a first software task where the first software task is written for the physical configuration,
> the array processor upon detection of the second state of the CSB operating in a second operating context adapted for a second software task where the second software task is written for a second array processor having a different physical configuration. (emphasis added)

Panwar does not teach and does not suggest "a physical configuration of at least two processing elements." Panwar's system is not an array processor which supports multiple

physical configurations. Panwar merely teaches a single processor that can be configured to operate as multiple virtual processors. As admitted in the Official Action, Panwar also does not teach and does not suggest "a processor state register storing a context status bit (CSB), the CSB having a first state and a second state, each processing element operating to detect the state of the CSB." The Official Action relies on Gordon for the general idea of storing status data in a state register which represents multiple states.

Gordon addresses a tomography processor system for processing data from a plurality of projections in essentially real time to produce a reconstructed image immediately following completion of a scan. Gordon, Abstract. Fig. 2 of Gordon is a block diagram showing the major sections of the Gordon processor system which includes a corrector 112, a convolver/interpolator 114, image reconstructor 116, and the like. As relied upon in the Official Action, a status register 930 shown in Gordon's Fig. 21 contains information indicating the busy or idle states of the various sections of the processor system. Gordon, col. 47, lines 14-16. The status register contains individual bits whose value corresponds to the state of each section of the processor system. One bit tracks the status of the corrector while another bit tracks the status of the convolver. Gordon, col. 47, lines 16-38. None of the bits in Gordon's status register, however, track the operating context of an array processor as presently claimed.

Combining the teachings of Panwar and Gordon as the Official Action suggests would not result in the presently claimed features. The Official Action suggests that combining the state register of Gordon with the virtual processors of Panwar would result in a register in Panwar that would hold bits which specify whether a particular processor is napping. However, unlike the

present invention, Panwar's system utilizes one processor operating as a plurality of virtual processors. Such a combination would merely result in a register in Panwar's system which tracks the state of Panwar's virtual processors without allowing a software task written for a different physical configuration to operate.

Panwar and Gordon, taken separately or in combination, do not teach and do not suggest "a physical configuration of at least two processing elements" as presently claimed in claim 14. Further, Panwar and Gordon, taken separately or in combination, do not teach and do not suggest "each processing element operating to detect the state of the CSB," as presently claimed in claim 14. Panwar and Gordon, taken separately or in combination, do not teach and do not suggest "the array processor upon detection of the first state of the CSB operating in a first operating context adapted for processing a first software task where the first software task is written for the physical configuration" as presently claimed in claim 14. (emphasis added) Panwar and Gordon, taken separately or in combination, do not teach and do not suggest "the array processor upon detection of the second state of the CSB operating in a second operating context adapted for a second software task where the second software task is written for a second array processor having a different physical configuration," as presently claimed in claim 14. (emphasis added) See also claim 19 which addresses a method version of apparatus claim 14.

With regards to claim 18, the Official Action suggests that "it is inherent that each processing element has a physical identifier." Applicants respectfully disagree. According to the MPEP Section 2112, "[t]o establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that

it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.' " The MPEP continues by further stating "[i]n relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art," citing *Ex parte Levy*, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990) (emphasis in original). The Examiner has not met this burden here.

As addressed above, Panwar merely utilizes multiple virtual processors on a single processor, the number of multiple virtual processors is dependent on the needs of the software application. Panwar, Col. 7, line 19. It cannot be assumed that each virtual processor of Panwar has both a virtual and physical identifier as claimed. In contrast to Panwar, claim 18 of the present invention recites "[a]n array processor comprising: a physical configuration of at least two processing elements .... wherein each processing element of the at least two processing elements has a physical identifier and a virtual identifier, wherein during the processing of the second software task, instructions are loaded to each processing element according to its virtual identifier." Simply put, the presently claimed approach is not taught and is not rendered obvious by Panwar and Gordon.

Claims 16 and 17 were rejected under 35 U.S.C. §103(a) based on Panwar and Gordon, as applied above, and further in view of Dowling. Since claims 16 and 17 depend from and

contain all the limitations of claim 14 as presently amended, claims 16 and 17 distinguish from the references in the same manner as claim 14.

Claims 24-27 and 29 were rejected under 35 U.S.C. §103(a) based on Panwar in view of Gordon, as applied above, and further in view of Keckler. The Official Action further relies on Panwar as purportedly teaching an apparatus for providing efficient sharing of programming resources in a merged very long instruction word (VLIW) sequence processor (SP) and VLIW processor element (PE) processor, the merged VLIW SP/PE processor configurable in a first merged processor configuration or in a second merged processor configuration. Applicants respectfully disagree. As described above, Panwar addresses a single processor operating as different number of virtual processors. Unlike the present invention, Panwar teaches a single physical configuration. It does not teach and does not suggest a merged VLIW SP and a VLIW PE processor which is configurable to operate in a first merged processor configuration or in a second merged processor configuration having a different physical configuration.

The present invention as claimed in claim 24 addresses a merged VLIW SP/PE processor which can operate in multiple processor configurations. To select either a first merged processor configuration or a second merged processor configuration, the merged VLIW SP/PE processor reads a context select bit (CSB) stored in a processor state register and a SP/PE-bit of a received instruction. Depending on the values of these two bits, the merged VLIW SP/PE processor determines how to operate in the first merged processor configuration or the second merged processor configuration. When operating in the first merged processor configuration, the merged VLIW SP/PE processor accesses at least one register from a second set of registers upon

processing an SP instruction. This advantageous approach allows a resource file which is typically allocated for use by parallel instructions executing in a PE may be utilized by a sequential instruction executing in an SP. Claim 24, as presently amended, recites "a processor state register storing a context select bit (CSB), the merged VLIW SP/PE processor reading the values of the CSB and the SP/PE-bit of an instruction to select a first merged processor configuration or a second merged processor configuration when processing the instruction, the first merged processor configuration adapted for accessing at least one register from the second set of registers when processing an SP instruction."

The Official Action further relies on Panwar at col. 14, lines 57-63 and Fig. 11 as purportedly teaching "an SP resource file having a first set of registers" and "a PE resource file having a second set of registers." Applicants respectfully disagree. At the cited portion of Panwar, an integration execution unit (IEU), besides its execution function, includes mechanisms to access single and double precision architectural registers. The IEU is neither an SP resource file nor a PE resource file as claimed. Furthermore, at the cited portion of Panwar, a copy of integer architectural register files is provided for each virtual processor. Panwar's disclosure is silent with respect to any further description of its an integer architectural register file. Unlike Panwar, an SP resource file and a PE resource file, as claimed in claims 24-27 and 29, are primarily allocated for use by an SP and a PE, respectively. A copy of integer architectural register files is different than separate resource files such as the presently claimed SP resource file and the PE resource file.

The Official Action relies on Keckler's Fig. 1 as purportedly teaching a system in which a VLIW comprises at least two instructions where the instructions may belong to different threads. Keckler addresses a parallel data processing system which utilizes very long instruction words (VLIWs) to achieve instruction level and thread level parallelism simultaneously. Keckler's parallel processors are shared by threads of computation and may process operations from different instruction words of different threads of computation such that the execution of threads is interleaved across the parallel processors. Keckler, col. 2, lines 1-20.

The Official Action than makes a logical leap by equating thread IDs and SP/PE bits when combining the teachings of Keckler with Panwar. Although Keckler does not specifically disclose a thread identifier (ID), it is known in the art that a thread ID refers to a group of instructions that execute under the same thread context. Unlike each instruction carrying a thread ID, claim 24 claims an SP/PE bit carried in each instruction. In the present invention, the combination of the SP/PE bit carried in an instruction and the value of the CSB provide the basis upon which the merged VLIW SP/PE processor selects the merged processor configuration when processing the instruction. Claim 24 recites "the merged VLIW SP/PE processor reading the values of the CSB and the SP/PE-bit of an instruction to select a first merged processor configuration or a second merged processor configuration when processing the instruction."

Panwar, Gordon, and Keckler, taken separately or in combination, do not teach and do not suggest "a processor state register storing a context select bit (CSB), the merged VLIW SP/PE processor reading the values of the CSB and the SP/PE-bit of an instruction to select a first merged processor configuration or a second merged processor configuration when

processing the instruction" as presently claimed in claim 24. Furthermore, Panwar, Gordon, and Keckler, taken separately or in combination, do not teach and do not suggest "the first merged processor configuration adapted for accessing at least one register from the second set of registers when processing an SP instruction," as presently claimed in claim 24.

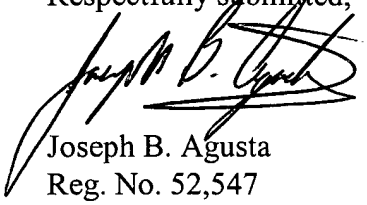
With regard to claim 25, the Official Action comments that a "PE instruction" is just a name of an instruction. Applicants respectfully disagree.

Claim 28 was rejected under 35 U.S.C. §103(a) based on Panwar in view of Gordon and Keckler, and further in view of Bapst. Since claim 28 depends from and contains all the limitations of claim 24 as presently amended, claim 28 distinguishes from the references in the same manner as claim 24.

Conclusion

All of the presently pending claims, as amended, appearing to define over the applied references, withdrawal of the present rejection and prompt allowance are requested.

Respectfully submitted,



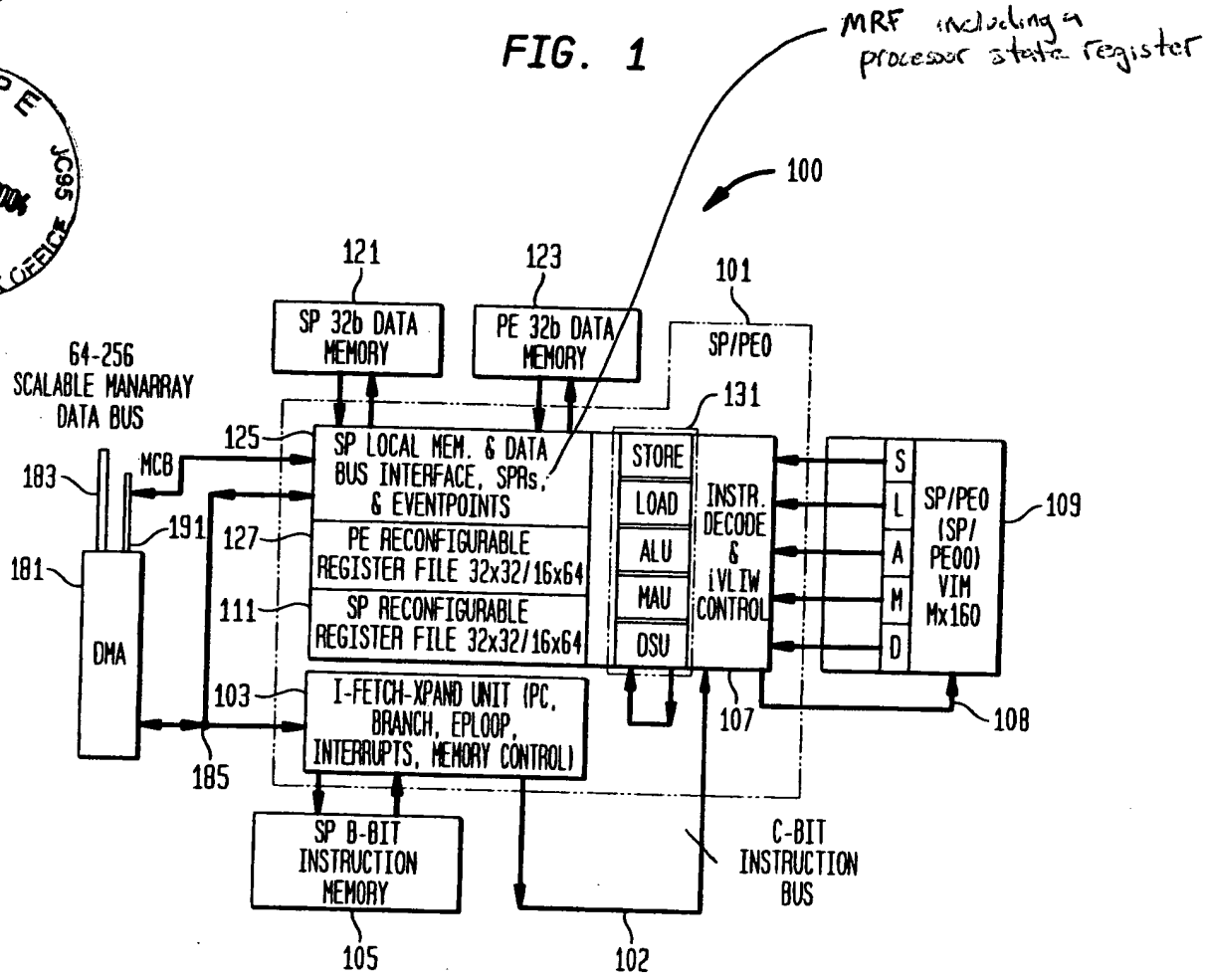
Joseph B. Agusta
Reg. No. 52,547
Priest & Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713-7736
(919) 806-1600

ANNOTATED SHEET SHOWING CHANGES FOR APPLICATION SERIAL NO. 10/761,564

SERIAL NO.: 09/598,558
PETER H. PRIEST (919-806-1600)

1/6

FIG. 1



ANNOTATED SHEET SHOWING CHANGES FOR APPLICATION SERIEL NO. 10/761,564

FIG. 5A